Association for Information Systems
AIS Electronic Library (AISeL)

Wirtschaftsinformatik Proceedings 2005 | Wirtschaftsinformatik

February 2005

Ontology Evolution: MEDLINE Case Study

Andreas Abecker
University of Karlsruhe

Ljiljana Stojanovic
University of Karlsruhe

In: Ferstl, Otto K, u.a. (Hg) 2005. *Wirtschaftsinformatik 2005: eEconomy, eGovernment, eSociety*; 7. Internationale Tagung Wirtschaftsinformatik 2005. Heidelberg: Physica-Verlag

ISBN: 3-7908-1574-8

Ontology Evolution: MEDLINE Case Study

Andreas Abecker, Ljiljana Stojanovic
University of Karlsruhe

Abstract: With the rising importance of knowledge interchange, many industrial and academic applications have adopted ontologies as their conceptual backbone. Business dynamics and changes in the operating environment often give rise to continuous changes in application requirements that may be fulfilled only by changing the underlying ontologies. This is especially true for Semantic Web applications, which are based on heterogeneous and highly distributed information resources and therefore need efficient mechanisms to cope with changes in the environment. In our previous work we have developed the KAON ontology evolution framework that (i) enables handling the required ontology changes; (ii) ensures the consistency of the underlying ontology and all dependent artefacts; (iii) supports the user to manage changes more easily; and (iv) offers advice to the user for continual ontology reengineering. In this paper we apply the proposed approach on the MEDLINE system and discuss its benefits. First, we translated the MeSH/MEDLINE into a set of the ontologies by enriching the MeSH vocabulary with the set of rules and by eliminating some inconsistencies. Second, we showed that ontology evolution ensures the consistency between all related data. Third, we indicated how formal semantics provided by an ontology might be useful to improve the indexing in the existing MEDLINE system.

Keywords: Ontology Evolution, MEDLINE

1 Introduction

An important characteristic of today's business systems is their ability to adapt themselves efficiently to the changes in their environment, as well as to the changes in their internal structures and processes. The continual reengineering of a business system, i.e. the need to be better and better, is becoming a prerequisite for surviving in the highly changing business world. Although changes encompass several dimensions of a business system (e.g. people, processes, technologies), most of them are reflected on its IT infrastructure. For example, the establishment of a new department in the organisational structure will require the corresponding changes in the enterprise portal, underlying groupware system, skill management system, etc. Therefore, the adaptability of the implemented IT solutions directly defines the efficiency of a business system.

However, building and maintaining long-living applications that will be "open for changes" is still a challenge for the entire software engineering community. Even though there are ongoing attempts to address this problem by providing IT systems with powerful concepts for self-management [KeCh03], they focus only on changes caused by malfunctioning of a system. Indeed, most of today's management tasks are performed manually. This can be time-consuming and error prone. Moreover, it requires a growing number of highly skilled personnel, making the maintenance of applications costly.

It is clear that an ad hoc management of changes in applications might work only for particular cases. Moreover, it can scale neither in space nor in time. Therefore, in order to avoid drawbacks in the long run, the change management must be treated in a more systematic way. It is especially important for the applications that are distributed over different systems. Examples of such applications are knowledge management applications that enable integration of various, physically distributed knowledge sources differing in the structure and the level of formality.

In order to avoid unnecessary complexity and possible failures and/or even to ensure the realisation of a request for a change, the *change management should deal with the conceptual model* of such an application. For example, a more efficient retrieval of knowledge items in a knowledge management system requires the establishment of the (hierarchical) relationships between their conceptual descriptions.

Ontologies have recently become a key technology for semantics-driven modelling, especially for the ever-increasing need for knowledge interchange and integration. Many industrial and academic applications have adopted ontologies as their conceptual backbone. The usage of ontologies has several advantages [Fen+03; UsGr96]:

- Ontologies facilitate *interoperability between applications* by capturing a shared understanding of a problem domain. They provide comfortable means for explicating implicit design decisions and underlying assumptions at the system building time. This makes it easier to reason about the intended meaning of the information interchanged between two systems.
- Ontologies provide a *formalization* of a shared understanding which makes them *machine-processable*. Machine processibility is the basis for the next generation of the WWW, the so-called *Semantic Web* [Bee00], which is based on using ontologies for enhancing (i.e. annotating) content with formal semantics.
- The explicit representation of the semantics of data through ontologies enables applications to provide a qualitatively *new level of services,* such as verification, justification, gap analysis, etc.

Ontology-based applications are subject to a continual change. Thus, to improve the speed and to reduce costs of their modification, the changes have to be re-

flected on the underlying ontology. Moreover, as ontologies grow in size, the complexity of change management increases significantly. If the underlying ontology is not up-to-date, then the reliability, accuracy and effectiveness of the system decrease significantly [KlFe01]. For example, an obsolete classification of knowledge items in an ontology-based knowledge management system decreases the precision of the knowledge retrieval process. A typical example is the MEDLINE system[1], the largest medical knowledge base available over the Internet, which is based on the MESH medical ontology. In order to stay in line with the state-of-the art in medical research, MESH is frequently updated. However, the ontology is not only extended with new terms (e.g. new diseases and medicaments); rather, the terms are often reclassified according to the latest research results. Therefore, in case the MESH is obsolete, not only that some relevant information will be missing, but also some wrong answers will be delivered.

Since an ontology has to be continually changed, the need for the ontology evolution[2] is inevitable. The task of the ontology evolution is to formally interpret all requests for changes coming from different sources (e.g. users, internal processes, business environment) and to perform them on the ontology and depending artefacts while keeping consistency of all of them. Figure 1 illustrates the role of the ontology evolution in a business system.

Indeed, ontology evolution is defined as the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artefacts [Sto04]. Since a change in the ontology can cause inconsistencies in other parts of the ontology, as well as in the dependent artefacts, the ontology evolution has to be considered as a process. It encompasses the set of activities, both technical and managerial, that ensures that the ontology continues to meet organizational objectives and users' needs in an efficient and effective way.

In our previous work [Mae+03] we proposed an approach for the evolution between dependent and distributed ontologies. The approach is based on the process model [Sto+02] that (i) enables handling the required ontology changes; (ii) ensures the consistency of the underlying ontology and all dependent artefacts; (iii) supports the user to manage changes more easily; and (iv) offers advice to the user for continual ontology reengineering. The proposed approach has been implemented in the KAON[3] ontology management system. In this paper we present the evaluation of the proposed approach on the MEDLINE dataset .

---

1 http://www.nlm.nih.gov/pubs/factsheets/medline.html

2 The word "evolution" merely means "change through time". It implies neither a direction, nor, necessarily, improvement, but merely a change.

3 kaon.semanticweb.org

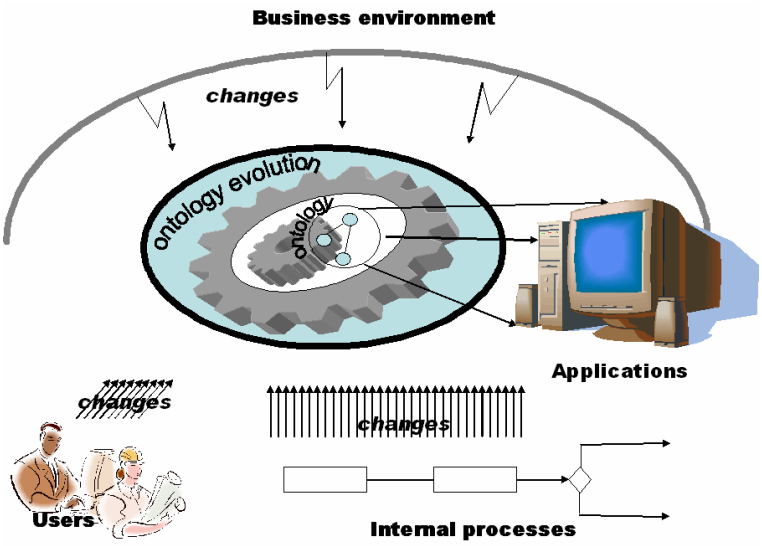

Figure 1: The role of ontology evolution in a business system

2 Case Study

To demonstrate the usefulness of our approach for the ontology evolution, we applied it to the MeSH[4] (MEdical Subject Headings). MeSH is a controlled vocabulary used for indexing medical documents. The goal of the MeSH is to provide a reproducible partition of concepts relevant to biomedicine for the purpose of organising knowledge and information. In biomedicine and related areas, new concepts are constantly emerging, old concepts are in a state of flux and terminology and usage are modified accordingly. To accommodate these changes, the MeSH has to be updated as well as the articles indexed by the MeSH. Indeed, the main reason for using the MeSH as a case study is that the National Library of Medicine (NLM) produces the MeSH with an annual update cycle. Since the MeSH is used in real medical systems, management of its change is a critical issue.

The NLM has produced the MEDLINE[5] database since 1966. The MEDLINE database includes over 10 million literature quotations of articles written in 41 languages. Each article is indexed with the MeSH descriptors assigned by an individual who reads the article in its original language and assigns the descriptors to indicate what the article is about. About 400.000 articles are indexed per year. The

[4] http://www.nlm.nih.gov/mesh/
[5] http://www.nlm.nih.gov/pubs/factsheets/medline.html

MeSH is now in its 40th year of production and is added to and otherwise modified on an annual basis. Beginning in 2002, over 2.000 completed references are added daily each Tuesday through Saturday, January through October (over 460,000 added last year). These modifications are then applied to the MEDLINE database; articles are not re-indexed, but the database is kept current with the current version of the MeSH. This is a time-consuming activity since two months (November and December) are needed to make the transition of the NLM to a new year of the MeSH vocabulary used to index the articles.

According to the official MeSH web site[6], the following changes are applied on the MeSH version from 2003:

- 666 descriptors were added representing topics with no directly corresponding descriptors in the MeSH version used in 2003. The most recent examples of such additions are "Severe Acute Respiratory Syndrome" and "SARS Virus";
- 109 descriptors were replaced with more up-to-date terminology;
- 20 descriptors were deleted;
- 484 terms were added.

The practical experiences with the MEDLINE show that it is easy to add something (either a descriptor to the MeSH or an indexed article to the MEDLINE), but it is hard to modify data that are already in the system. The authors of the MEDLINE system found out that meaning of change is important and that there is a need for an update model [Nel01].

The goal of the MeSH/MEDLINE case study was to show that:

- our ontology evolution system is able to work with large ontologies such as the MeSH. The newest version of the MeSH (MeSH 2004) contains 22.568 descriptors, 83 qualifiers and 137.557 supplementary concept records. The meaning of the MeSH entities is described in next subsection;
- the dependent/distributed ontology evolution [Mae+03] might be applied on the MEDLINE since the MeSH itself consists of several independent parts and the medical articles are only annotated by the MeSH;
- formal semantics provided by an ontology might be useful to improve the indexing in the existing MEDLINE system.

Our work regarding the MeSH can be split into three phases:

- Phase 1 – the representation of the MeSH in the form of the KAON ontologies [Mot02];

[6] http://www.nlm.nih.gov/pubs/techbull/nd03/nd03_mesh.html

- Phase 2 – the evaluation of the applicability of the ontology evolution support on the MeSH/MEDLINE;
- Phase 3 – the suggestions for the continual improvement of the MEDLINE.

These phases are subsequently described.

2.1 Phase 1

Our first task was to transfer all information available in the MeSH repository into the KAON system in order to verify whether our ontology evolution system can be used at all. This required (i) the understanding the MeSH and (ii) the creation of the KAON ontologies that mimic the MeSH.

Understanding the MeSH requires an understanding of its structure. There are three major components to the MeSH:

- descriptors;
- subheadings (also known as Qualifiers);
- supplementary concepts.

Descriptors (e.g. "*Headache*") are the main headings. Qualifiers (e.g. "*Therapy*", "*Diagnosis*", etc.) are used with descriptors and afford a means of grouping together the documents concerned with a particular aspect of a subject. Indeed, qualifiers are used to modify (refine) descriptors by indicating particular aspects. They are used in indexing, cataloguing, and online searching to qualify the MeSH descriptors by pinpointing some specific aspect of the concept represented by the descriptor. For example, "*LIVER/drug effects*" indicates that the article or book is not about the liver in general but about the effect of drugs on the liver. Supplemental (e.g. "*Ametohepazone*") is added daily and is largely chemicals.

The MeSH structure is centred on descriptors, concepts, and terms [Nel+01]. A descriptor is viewed as a class of concepts, and a concept as a class of synonymous terms within a descriptor class. Indeed, a descriptor class consists of one or more concepts closely related to each other in meaning. For example, for the "*Headache*" descriptor the concepts "*Headache*" and "*Sharp Headache*" are defined. For the purposes of indexing, retrieval, and organisation of the literature, these concepts are best lumped together in one class. Each descriptor has a preferred concept. Further, one of the terms naming that concept is the preferred term of the preferred concept, and takes on the role of naming the descriptor. Each of the subordinate concepts also has a preferred term, as well as a labelled (broader, narrower, related) relationship to the preferred concept. Terms meaning the same are grouped in the same concept. For the previously mentioned descriptor "*Headache*" following terms among others are defined "*Head Pains*", "*Head-Pain*", "*Cephalgias*".

An example is shown in Figure 2. It can be seen that concept classes II and III are respectively, narrower and related to concept class I (the preferred concept), but are not equivalent to each other. Each concept class could be given its own definition if desired. It can also be seen that "*HIV Encephalopathy*" and "*AIDS Encephalopathy*" are synonymous terms within the same concept class.

Relationships among concepts can be represented explicitly in the thesaurus, most notably as relationships within the descriptor class. Hierarchical relationships are represented as broader or narrower (parent-child) relationships between concepts within descriptors. Other types of relationships include associative relationships such as the Pharmacological Actions or see-related cross-references as well as forbidden combination expressions such as the Entry Combination. For example, the MeSH concept "*Headache*" is broader than the MeSH concept "*Bilateral Headache*", the MeSH concept "*Sharp Headache*" is narrower than the MeSH concept "*Head Pains*" or the MeSH concepts "*Headache*" and "*Head Pains*" are related.

AIDS DEMENTIA COMPLEX [Descriptor Class]
Concept Class I - Preferred Concept
Terms: AIDS Dementia Complex (Preferred Term)
HIV Dementia
HIV-Associated Cognitive Motor Complex
Dementia Complex, AIDS-Related
Concept Class II - Subordinate Concept (narrower)
Terms: HIV Encephalopathy (Preferred Term)
AIDS Encephalopathy
Concept Class III - Subordinate Concept (related)
Terms: HIV-1-Associated Cognitive Motor Complex (Preferred Term)

Figure 2: An example of the MeSH descriptors

Three kinds of informative references may be found in descriptor records: "*see related*", "*consider also*", and "*main heading/subheading combination*" references. "*See related*" references indicate the presence of other descriptors that are conceptually related to the topic. The "*consider also*" notation is primarily used on anatomical descriptors. The "*main heading/subheading combination*" notations refer an invalid (and prevented) combination of descriptors.

Based on the analysis of the MeSH structure, we develop several ontologies. They are shown in Figure 3. The goal was to model all information that exists in the MeSH model including the implicit knowledge. Therefore, the approach can be summarised as follows:

- the model of the MeSH is transformed into the MeSH ontologies;

- "hidden" (hard-coded) knowledge embedded in the MeSH is translated into a set of rules in the corresponding ontologies and is used in typical inferencing tasks.

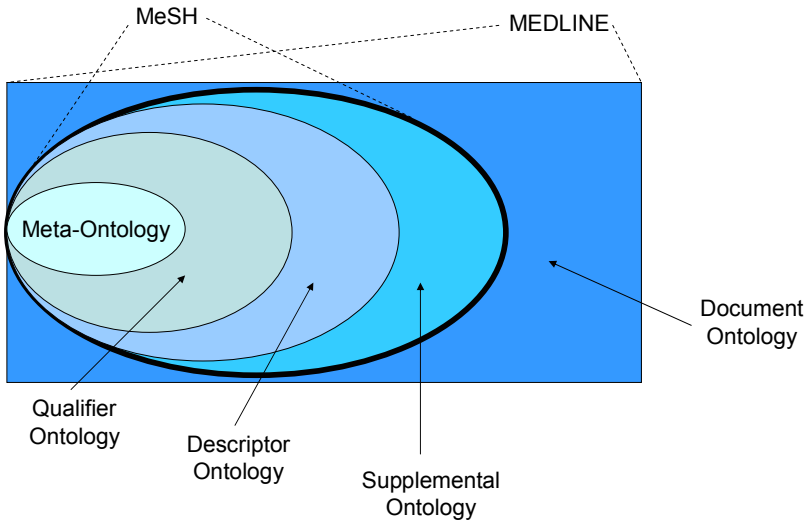

Figure 3: Representation of the MeSH and the MEDLINE as KAON ontologies

The *Meta-Ontology* shown in Figure 4 represents the conceptual model of the MeSH. It contains concepts such as "*Descriptor*", "*Qualifier*", "*Supplemental*", "*Concept*", "*Term*", etc. The relationships between them are defined according to the MeSH model. We extend this model by representing explicit all information that was implicit in the MeSH model. The semantic of the MeSH model is implicit, hidden in the XML files and difficult to discover. By providing explicit semantics of the MeSH relationships, it is possible to perform the formal verification of a model. Such an approach is described in section 2.3.

Actually, in an ontology there are two types of implicit knowledge: the *axioms* and *general rules*. *Axioms* are the standard set of rules such as rules for symmetric, transitive and inverse properties. For example, if A "*is related to*" B, B "*is related to*" C, and "*is related to*" is a transitive property, then the ontology system can infer that A "*is related to*" C as well. Thus, we do not need to express this information explicitly. *General rules* are domain specific rules that are needed to combine and to adapt information available in the ontology. They are used to specify the relationship between ontology entities in the form of rules. For example, if C "*is preferred* ***concept*** *for*" D and T "*is preferred* ***term*** *for*" C, then it can be concluded that T "*is preferred* ***term*** *for*" D.

In general, axioms and rules are used to infer new knowledge. The possibility to derive information makes the model of a domain more concise, more accurate, and easier for maintenance. Obtaining and formalising the non-explicit but available knowledge about the knowledge model of the MeSH ensures the advantages over other medical systems.

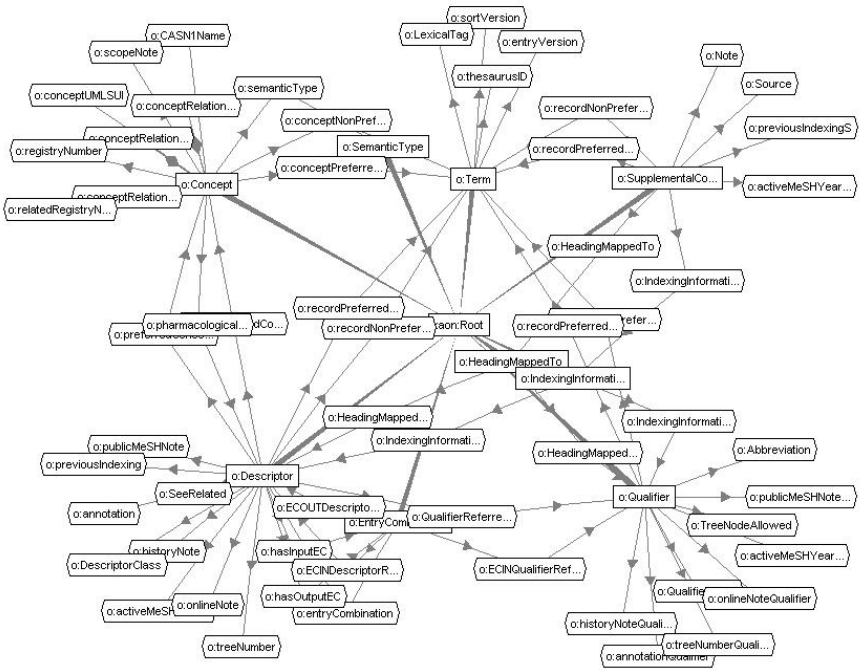

Figure 4: The Meta-Ontology representing the conceptual model of the MeSH

The implicit knowledge of the MeSH is explicitly modelled through rules. For example, for the "*see related*" relationship, the symmetry axiom may be exploited when searching for information. Without the definition of this axiom, searching for related descriptors might depend on the way metadata was provided for them. If one defines that some descriptor named "*X*" has the "*see related*" relationship with some other descriptor named "*Y*", there is no possibility (without programming or explicit specification) to find out that the descriptor "*Y*" also has the "*see related*" relationship with the descriptor "*X*". Further, it is impossible to conclude that the descriptors "*X*" and "*Y*" cannot be in the "*main heading/subheading combination*" relationship to each other.

The *Qualifier ontology* is based on the Meta-Ontology since it defines the structure of the MeSH/MEDLINE system. It contains all the MeSH qualifiers that are

represented as subconcepts of a concept "*Qualifier*" that is defined in the Meta-Ontology.

The *Descriptor ontology* contains information about the concrete MeSH descriptors. Therefore, it also reuses the Meta-Ontology. Further since one descriptor may reference to the qualifier concepts, the Descriptor ontology includes the Qualifier ontology as well. For example, the descriptor "*Calcimycin*" has a reference to the qualifier "*abnormalities*".

The *Supplemental ontology* reuses the Qualifier and the Descriptor ontologies directly and indirectly the Meta-Ontology through both of the directly included ontologies. It specialises the concept "*Supplemental*" defined in the Meta-Ontology according to the MeSH context. Moreover, it establishes the reference between qualifiers, descriptors and supplemental concepts.

The MeSH is used for indexing biomedical articles. This information is stored in the MEDLINE. Each index (or annotation in the Semantic Web terminology) consists of the MeSH *headings* and *chemicals*. Each MeSH *heading* contains one pair or more pairs of descriptors and qualifiers. Each pair defines a main topic of the article and is considered as a whole. On the other hand, *chemical* contains a supplemental concept that describes more specific topics of an article. To model this information we have developed the so-called *Document ontology*. It contains only the metadata about biomedical articles and not the articles themselves. It includes all the previously mentioned ontologies. We note that we transfer all information about the MeSH but only about 100.000 indexed documents. The reasons for selected the MEDLINE subdomain are discussed later.

2.2 Phase 2

In the second phase we evaluate the possibility to apply our ontology evolution system to the set of the ontologies generated from the MeSH/MEDLINE system. It is worth noting that we cannot compare our system with the existing MEDLINE system due to two reasons:

1. There is no MEDLINE maintenance system that enables keeping consistency. For example, after removal of some descriptor from the MeSH, it might be possible that some articles are still indexed with the descriptor that does not exist any more. All changes are performed manually. Thus, any modification is a time-consuming and error-prone activity;
2. The MeSH is available on the Internet NLM home page at http://www.nlm.nih.gov/mesh/filelist.html. However, the MEDLINE[7] can be searched free of charge but access to the MEDLINE services is provided by organisations that lease the database from NLM. Therefore, we were not able to work

[7] http://www.nlm.nih.gov

with the full content of the MEDLINE. We manually downloaded about 100.000 articles and their annotation by making query about "*Headache*" and parsing the XML output. Even though this restriction is made, the evaluation results are applicable.

Moreover, there are not other case studies that deal with large-scale information systems and that use formal ontologies. The comparion of the functionality of the existing ontology evolution systems is given in [Sto04$^+$].

As already mentioned, the goal of this phase is to demonstrate that the evolution of medical vocabulary can be automated. The first application of our ontology evolution system was during the creation of the KAON version of MeSH/MEDLINE ontologies. We found several anomalies (such as redundancies, inconsistencies and undefined entities) in the existing MeSH/MEDLINE data. For example, several descriptors were defined twice. Moreover, in the XML file each reference is stored through two elements: entity ID and entity names. Since one entity is referenced in several entities, different names are used for the same entity. Note that synonyms are represented as terms. Finally, we found references to the undefined entities. This problem may be a consequence of a syntax error in the XML file or may be a consequence of the manual change propagation procedure since the people might not find all effects of a change.

Since the ontology evolution system was applied during the creation of the MeSH/MEDLINE ontologies, all these anomalies were prevented. Here we show how the consistency can be enforced when the initial consistent ontologies already exist.

Therefore, the result of the first application of the ontology evolution system is a set of consistent MeSH/MEDLINE ontologies. Then, we try to modify these ontologies using our ontology evolution system. We decide to modify the Descriptor Ontology since descriptors are created for the purpose of indexing the medical literature. Since the worst case is the concept deletion, we measured time needed to perform this change and the number of generated changes. Note that there is no goal system that can be used for comparison. In the MEDLINE system the semantics of change[8] as well as the change propagation[9] are performed manually. Therefore, we only wanted to show that the removal could be performed in acceptable time, which is much faster and more accurate than in the existing system. Since we selected the subdomain of "*Headache*" diseases and included articles about this topic and their annotation into the Document Ontology, the descriptor "*Head-*

[8] The semantics of the change phase of the ontology evolution process prevents inconsistencies by computing the additional changes that guarantee the transition of the ontology into a consistent state.

[9] The task of the change propagation phase of the ontology evolution process is to bring automatically all dependent artefacts (i.e. ontology instances on the Web, dependent ontologies and application programmes using the changed ontology) into a consistent state after an ontology update has been performed.

ache" is chosen for removal. It is represented as the concept "*Headache*" in the Descriptor ontology. It is visible in all the ontologies that reuse the Descriptor ontology. Thus, the request for the removal of the concept "*Headache*" might have consequences on the Supplemental Ontology and the Document Ontology as well (see Figure 3). Consequently, the dependent ontology evolution[10] has to be applied since the synchronisation between the Document ontology and the ontologies that include it is necessary for a consistent and, therefore, efficient, effective and accurate system.

Note that the Meta, the Qualifier, the Descriptor and the Supplemental ontologies are stored within one ontology server and the Document ontology is stored on a separate ontology server. Thus, by changing the Descriptor ontology, we were able to apply all ontology evolution "types"[11]. The single ontology evolution is applied to the Descriptor ontology, the dependent ontology evolution is applied to the Supplemental ontology since it reuses the Descriptor ontology through the inclusion while the distributed ontology evolution is applied to the Document ontology since it reuses the Supplemental ontology through the replication. Consequently, the results that are obtained for the Descriptor ontology and the Supplemental ontology are completely correct whereas the results obtained for the Document ontology are only the approximation since this ontology contains only a part of all MEDLINE articles.

Even though the Document ontology contains 2.417.584 entities, our ontology evolution system was able to perform the deletion of the concept "*Headache*" in this ontology in 218 seconds[12]. The removal of that concept in the Supplemental ontology (that includes the Document ontology as well) lasted about 50 seconds longer since there are not so many entities in the Supplemental ontology that have reference to the concept "*Headache*" from the Descriptor ontology. The removal in the Document ontology took 1.583 seconds since almost all documents are annotated. Note that the complexity of the dependent ontology evolution depends on the number of instances in a linear way. Therefore, the existence of more instances (i.e. annotated articles) will linearly increase the time needed to perform a change.

The following set of additional changes was generated:

---

10 A dependent ontology is an ontology that includes ontologies located at the same node on the network. A distributed ontology is an ontology that includes ontologies located at different nodes on the network.

11 We identified two dimensions of the overall ontology evolution problem [Mae+03]. The first dimension defines the number of the ontologies that have to be updated for a change request. The second dimension specifies the physical location of evolved ontologies. Since it is not possible to fragment one ontology across many nodes, ontology evolution can be discussed at three levels: (i) evolution of the single ontology; (ii) evolution of dependent ontologies; (iii) evolution of the distributed ontologies.

12 We note that any change in the existing MEDLINE system is a time-consuming and error-prone activity, since it must be performed manually.

- 58 changes in the Document ontology;
- 13 changes in the Supplemental ontology;
- more than 100.000 changes in the Document ontology.

The changes in the Document ontology cover the removal of properties defined for the concept "*Headache*" and their consequences. Moreover, there are several subconcepts of the "*Entry Combination*" concept that establish the reference between the descriptor headache and corresponding qualifiers. All of them have to be removed as well. In the Supplemental ontology the descriptors are referenced through the property "*hasReferencedDescriptor*" and its specialisation. Therefore, the request for the removal of the concept "*Headache*" in the Document ontology requires the removal of this concept from the range of all these properties. Finally, all the annotated articles were about headache. Therefore, the annotation of all of them must be updated.

We believe that the usability of the MEDLINE management system might be significantly improved by incorporating the KAON ontology evolution approach [Mae+03; Sto+02; Sto04]. It does not only guarantee consistency. Rather, it improves the usability of the system by informing the responsible persons about all the consequences of a change since only in that way would they be able to comprehend the impact of a change and undo the unnecessary changes. In the next section we discuss the way in which the formal semantics provided by an ontology can be further exploited.

2.3 Phase 3

The assignment of MeSH topics to articles of the MEDLINE system represents the state-of-the-art in human indexing. The professional indexers who perform this task have been trained for at least 1 year. Ten to twelve topics in the form Descriptor/Qualifier are associated to each article. Although such annotations help in searching for articles, the MEDLINE suffers from information overloading. For example, searching the MEDLINE using the MeSH topic "*common cold*"[13] yields over 1,400 articles written in the last 30 years. Finding a relevant article might take 20-30 minutes.

We applied the data-driven change discovery[14] [Sto04] to improve annotations in the MEDLINE, since they are made manually. Since we assume that an annotation must be consistent with the underlying MeSH system, the "quality" of the annotation is assessed through the existence of redundancy, inaccurate or incomplete information. Note that we assume that the annotations are valid, i.e. all the metadata

[13] The example is taken from http://www.ovid.com.

[14] *The data-driven change discovery* considers the ontology instances in order to refine the ontology (including its instances as well).

in the annotation is consistent with the MeSH ontologies. This is guaranteed by applying the dependent ontology evolution as described in the previous section, which provides support for finding inconsistencies and resolving them.

Three quality criteria are defined in the following way:

- *Compactness* – A semantic annotation[15] is not compact or it is redundant if it contains more metadata than it is needed and desired to express the same "idea". In order to achieve compactness (and thus to avoid redundancy), the annotation has to comprise the minimal set[16] of the metadata without exceeding what is necessary or useful. The repetition of the metadata or the usage of several metadata with the same meaning only complicates maintenance and decreases the system performance;
- *Completeness* – An annotation is incomplete if it is possible to extend the annotation only by analysing the existing metadata in the annotation in order to clarify its semantics. It means that the annotation is not finished yet and requires that some additional metadata have to be filled in;
- *Aggregation* – An annotation is aggregative if it contains a set of metadata that can be replaced with semantically related metadata in order to achieve a shortened annotation, but without producing any retrieval other than the original annotation.

Note that assessment is performed on the annotation level and that the MeSH structure (i.e. a set of the MeSH ontologies) is the basis for all measures. This assessment can help refine and improve the annotation in the MEDLINE.

In order to clarify the meaning of the criteria here we give a short example that simulates the real MEDLINE system. It is shown in Figure 5.

2.3.1 Compactness

The concept hierarchy and the property hierarchy from the domain ontology are used to check this criterion. The first example in Figure 5 represents the incompact annotation because the article is annotated, after all, with the concept "*Person*" and its subconcept "*Female*". When someone searches for all articles about "*Person*", she searches for the articles about all its subconcepts (including "*Female*") as well. Consequently, she gets this article (minimum) twice. Moreover, such annotation introduces an ambiguity in the understanding of the content of an article, which implies problems in knowledge sharing. Let us examine the meaning of the annotation of a medical document using the set of metadata "*Person*", "*Female*",

---

15 An annotation consists of a set of ontology instances. We use term metadata as a synonym for an ontology instance.

16 An annotation is not minimal if excluding metadata results in the same retrieval for the same query, i.e. if precision and recall remain the same.

"*Aspirin*" and "*Complications*". Does it mean that the article is about complications in using aspirin only in females, or in all persons? When the second answer is the right one, then this article is also relevant for the treatment of male persons with aspirin. This implies new questions: is the annotation using metadata "*Female*" an error, or the metadata "*Male*" is missing? Anyway, there is an ambiguity in annotations, which can be detected and resolved by using our approach.

In order to prevent this, an article should be annotated using as special metadata as possible (i.e. more specialised sub-concepts). In this way, the mentioned ambiguities are avoided. Moreover, the maintenance of the annotations is also alleviated because the annotation is more concise and because only the changes linked to the concept "*Female*" (first example in Figure 5) can provoke changes in the annotation.

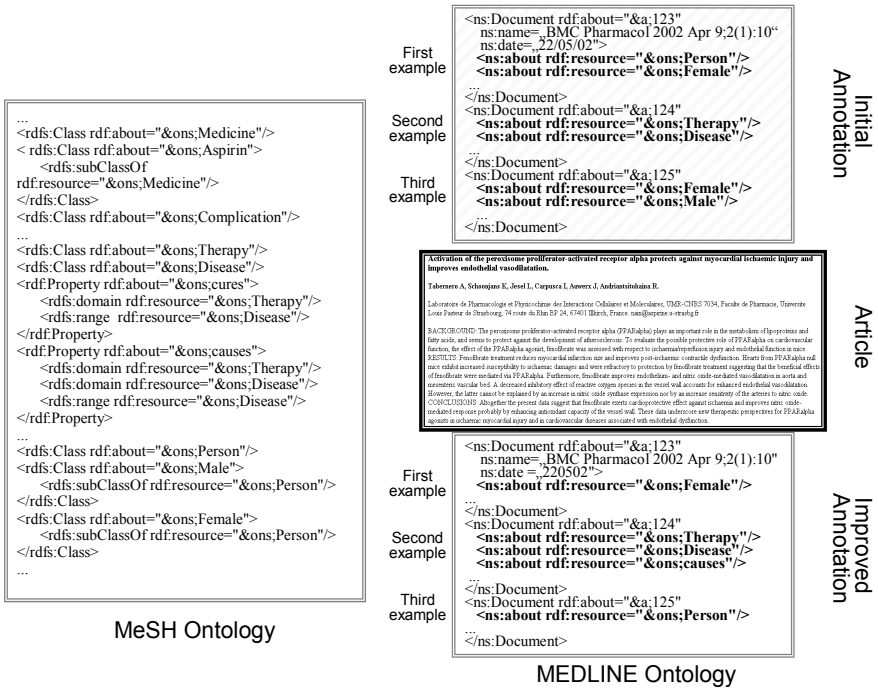

Figure 5: Annotation refinement based on the analysis the ontology structure and the existing annotations. The ontology is depicted in the left part. The right part shows downward the initial annotation, corresponding articles and the improved semantic annotation

2.3.2 Completeness

This criterion is computed based on the structure of the ontology. For example, one criterion is the existence of a dependency in the domain ontology between the domain entities, which are already used in the annotation. The second example in

Figure 5 contains concepts with many relationships between them (e.g. properties "*cures*" and "*causes*" exist between concepts "*Therapy*" and "*Disease*"). The interpretation is ambiguous since it is a question whether the articles are about how a disease (i) can be cured by a therapy, or (ii) caused by a therapy. In order to constrain the set of possible interpretations, the annotation has to be extended with one of these properties.

This problem is especially important when the repository of articles contains a lot of articles annotated with the same concepts because the search for knowledge retrieves irrelevant articles that use certain concepts in a different context. Consequently, the precision of the system is decreased.

2.3.3 Aggregation

This pattern for the annotation refinement occurs when an article is described with all subconcepts of one concept (e.g. concepts "*Female*" and "*Male*" as shown in the third example Figure 5). From the searching for articles point of view, it is the same whether an article is annotated using the combination of the concepts (e.g. "*Female*" and "*Male*") or using only the parent concept (e.g. "*Person*"). It is obvious that the second case of annotation makes the management much easier. Moreover, since the standard approaches to the ranking results of querying [Sto^{+}01] exploit conceptual hierarchies, for example in a querying for persons an article annotated using "*Female*" and "*Male*" will be placed at the same level as an article annotated using only one of these concepts. However, it has to be ranked on the top level (level of the concept "*Person*") because it covers all subtypes of the concept "*Person*".

3 Conclusion

Due to the ever increasing complexity, heterogeneity and physical distribution of the business, the importance of ontologies for the conceptualisation of the business applications becomes inevitable. It is especially important for the recently increased research in the Semantic Web and Web Services that enable publishing business processes on the Web.

However, the frequently changing business context implies the need to cope with changes in ontology-based business applications in a more systematic way. Firstly, different causes of changes (e.g. changes in the business environment, user's preferences, internal processes, etc.) have to be uniformly represented, in order to enable their efficient processing. Secondly, the changes have to be consistently resolved in the application, and their effects have to be propagated to all dependent business systems. Moreover, in order to control the resolution of the changes (e.g. the identification and overcoming of undesired changes), the responsible persons

have to be able to make appropriate decisions. Finally, the continual business re-engineering requires an automatic discovery of new changes by analysing the manner in which the application is used (e.g. the detection of trends in the users' behaviour). In order to fulfil these requirements efficiently, the managing of the changes in the ontology-based application has to be performed on the level of ontologies themselves. Therefore, the need for an efficient approach to the management of the changes in an ontology (e.g. ontology evolution) is obvious.

Moving ontologies into a large real-world context requires the scalability of the platforms they are dealing with. This is probably the most critical issue in the whole research related to ontologies. Can the approaches scale when their application data increases drastically? In the Semantic Web environment, such a data explosion is inevitable. We did our best in tackling the complexity problem in the KAON ontology evolution system. In this paper, we presented the MEDLINE evaluation study that shows the applicability of the KAON ontology evolution approach on the large datasets. Moreover, we illustrated how formal semantics provided by ontologies can be used to improve the indexing in the existing MEDLINE system. In that way, our approach goes beyond a standard change management process; rather it is a continual improvement process.

Acknowledgement

The research presented in this was partially financed by EU in the project "IST PROJECT 507237 - OntoGov".

References

[Bee00] Berners-Lee, T.: XML 2000 – Semantic Web talk, http//www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html, 2000.

[Fen+03] Fensel, D.; Hendler, J.A.; Lieberman, H.; Wahlster W. (Eds.): Spinning the Semantic Web: Bringing the World Wide Web to its full potential, MIT Press 2003, ISBN 0-262-06232-1

[Har+00] Hardless, C.; Lindgren, R.; Nulden, U.; Pessi K.: The evolution of knowledge management system need to be managed, Journal of Knowledge Management Practice, Volume 3, 2000.

[KeCh03] Kephart, J.; Chess, D.: The Vision of Autonomic Computing, IEEE Computer, January 2003, pp. 41-50.

[KlFe01] Klein, M.; Fensel, D.: Ontology versioning for the Semantic Web, In Proceedings of the 1st International Semantic Web Working Symposium (SWWS), Stanford University, California, USA, 2001. pp. 75-91.

[Mae+03] Maedche, A.; Motik, B.; Stojanovic, L.: Managing multiple and distributed ontologies on the Semantic Web, the VLDB Journal (2003) - Special Issue on Semantic Web, 2003, 12:286-302.

[Mot02] Motik, B.; Maedche, A.; Volz, R.: A conceptual modelling approach for building semantics-driven enterprise applications, In Proceedings of the First International Conference on Ontologies, Databases and Application of Semantics (ODBASE-2002), Springer, California, USA, LNCS 2519, 2002, pp. 1082-1099.

[Nel01] Nelson, S.: MeSH, UMLS, and the Semantic Web, Presentations at the Medical Information Society of Taiwan (MIST), Taoyuan, Taiwan, http://www.nlm.nih.gov/mesh/presentations/taiwan2001/semanticweb/index.htm, 2001.

[Nel+01] Nelson, S.; Johnston W.; Humphreys, B.: Relationships in Medical Subject Headings, Relationships in the organization of knowledge, edited by C.Bean and R. Green, Kluwer Academic Publishers, ISBN 0-7923-6813-4, 2001, pp.171-184.

[Sto+01]N. Stojanovic, A. Maedche, S. Staab, R. Studer, Y. Sure, SEAL - a framework for developing SEmantic portALs, In Proceedings of the international Conference on Knowledge Capture (K-CAP'01), Victoria, British Columbia, Canada , 2001., pp. 155-162.

[Sto+02] Stojanovic, L.; Maedche, A.; Motik, B.; Stojanovic, N.: User-driven Ontology Evolution Management, In Proc. of the 13th European Conference on Knowledge Engineering and Knowledge Management (EKAW'02), Siguenza, Spain, pp. 285-300, 2002.

[Sto04] Stojanovic, L.; An approach for continual ontology improvement, in Proceedings of the First International Conference on Knowledge Engineering and Decision Support (ICKEDS'2004), Porto, Portugal, 2004.

[Sto04+] Stojanovic, L.; Methods and tools for Ontology Evolution, PhD Thesis, University of Karlsruhe, 2004.

[UsGr96] Uschold, M.; Gruninger, M.: Ontologies: principles, methods, and applications, Knowledge Engineering Review, Volume 11, Number 2, 1996, pp. 93-155.